REMARKS

Applicants have considered the Final Office Action and the subsequent Decision on Appeal dated March 19, 2009, and the references cited therein. Claims 1-9 were previously pending. No claims have been allowed. Applicants have amended claims 1 and 8 to address the grounds for rejection set forth in the Decision on Appeal. Applicants have also added new claims 16-19 that unequivocally recite that the current and subsequent memory lines are *adjacent to the boundary* and the prefetch operation involves exactly a *single memory line*.

The Final Office Action from which the previous appeal was taken rejected the previously pending claims 1-9. In particular, claims 1, 4 and 6-8 were rejected under 35 U.S.C. Section 102(b) as being anticipated by Miller et al., U.S. Pat. No. 5,819,058 (Miller). Applicants limited their previous arguments on appeal to the grounds for rejecting claims 1 and 8. Claims 2, 3 and 9 were rejected under 35 U.S.C. Section 103(a) as being obvious over Miller in view of Mohamed et al., U.S. Pat. No. 6,684,319 (Mohamed). Claim 5 was rejected under 35 U.S.C. Section 103(a) as being obvious over Miller in view of Keller et al., U.S. Pat. No. 6,546478 (Keller).

Applicants have amended the previously rejected independent claims by incorporating claim 2 into claim 1 and incorporating a narrower version of claim 9 into claim 8. Thus, at least claim 8 differs in scope from the previously appealed claims.

The combined teachings of Miller and Mohamed neither disclose nor suggest each of the recited elements of independent claims 1 or claim 8 and, by definition, the elements of dependent claims 2-7 and 16-19. Applicants provide more particular reasons for traversing the current rejection herein below.

Applicants request favorable reconsideration of the Final Office Action's grounds for rejecting the previously pending claims in view of Applicants' remarks provided herein below. Please charge any fee deficiencies to Deposit Account No. 12-1216.

Applicants' Grounds for Traversing the Final Rejection of the Claims

Applicants specifically traverse the previous rejection of previously pending **claims 2, 3 and 9** as obvious over Miller in view of Mohamed. The amendments to **claim 1** include, among other things, incorporating elements of previously pending **claim 2** (now canceled). The

amendments to **claim 8** include, among other things, incorporating part of the recited claim elements in previously pending **claim 9**. Claims 2 and 9 have now been canceled.

Applicants' Invention Recited in Claim 1 (formerly claim 2)

Applicants' invention relates to a computer system with a processing unit and a memory. *See,* FIG. 3. The processing unit is arranged to fetch memory lines from the memory and execute instructions from the memory lines. *See*, FIG. 2, page 9, lines 29-31. Each memory line is fetched as a whole and being capable of holding more than one instruction.

Moreover, at least one instruction, from the memory lines comprising information, inserted at compile time, signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line. *See*, FIG. 2. Examples include stall, realign and prefetch bits. *See*, page 11, lines 25-33. The processing unit responds to the aforementioned information by controlling the part of processing (as signaled by the information).

With regard to the portion of claim 1 incorporated from now-canceled claim 2, the information inserted at compile time explicitly signals whether or not the subsequent memory line has to be prefetched during processing of the instruction, and the processing unit starts prefetching of the subsequent memory line in response to the information. *See*, FIG. 4, steps 44-49, page 15, lines 3-31.

Amended method claim 8 is supported by substantially the same portions of the disclosure recited above with regard to claim 1.

The Cited Prior Art

1. Miller

Miller discloses a system and method for compressing and decompressing variable length instructions contained in variable length instruction packets in a processor having a plurality of processing units. Miller discloses a compression system that includes a component for generating an instruction packet containing a plurality of instructions, a component for assigning a compressed instruction having a predetermined length to an instruction within the instruction packet. Miller's system also discloses a shorter compressed instruction corresponding to a more frequently used instruction. Miller also discloses a component for generating an instruction packet containing compressed instructions for corresponding ones of the plurality of processing units.

Furthermore, Miller discloses a decompression system that includes a component for storing a plurality of the instruction packets in a plurality of storage locations. A further component generates an address that points to a selected variable length instruction packet in the storage system, and a decompression component that decompresses the compressed instructions in the selected instruction packet to generate a variable length instruction for each of the processing units. The decompression system may also have a component for routing the variable length instructions from the decompression system to each of said processing units. *See,* Miller Abstract.

Miller's execution control unit ECU 26 controls retrieval and execution of instructions within the VLIW processor. *See*, column 4, lines 1-2, and column 8, lines 22-24. The compressed instructions comprise a stop bit (or an end of packet EP/NEP bit) that indicates to the ECU unit and processor where one compressed instruction packet ends and the next compressed instruction packet begins. *See*, column 5, lines 34-38. Accordingly the ECU 26 can calculate the length of the present instruction from the end of packet indicator. The ECU 26 can also calculate the start address of the next instruction. Miller uses a dual memory and a particular alignment so that the following instruction can be fetched in a single cycle. *See*, column 10, line 36 to column 11, line 2. Thus, a complete instruction packet is always stored in a single address of the first memory and the second memory of the dual memory. Furthermore, the addresses for the memory banks are generated separately as Aleft and Aright, so that the addresses can be independently incremented.

Miller states "However, in this system, the first and second memories are addressed in parallel to retrieve 128 bits so that instruction packets may cross the boundary between the memories." *See*, Column 10, line. 67 to column 11, line. 2.

2. Mohamed

Mohamed discloses a VLIW processor having a prefetch instruction buffer in addition to a cache. Mohamed discloses a decoder unit that is capable of executing a prefetch instruction. The decoder unit decodes a prefetch instruction flag bit that indicates when instructions are to be prefetched and placed into the buffer instead of into the cache. *See*, column 3, lines 35-39. The decoder unit signals a control unit, which sends the instruction code from a memory unit to the buffer and maintains an address mapping table and a program counter. The control unit also sets a select input on a multiplexer to indicate that the multiplexer is to output the prefetch

instructions it receives from the buffer. The multiplexer outputs the prefetch instructions to an instruction register that sends the prefetch instructions to appropriate functional units for execution.

3. Keller

Keller discloses a line predictor that caches alignment information for instructions. In response to each fetch address, the line predictor provides alignment information for the instruction beginning at the fetch address, as well as one or more additional instructions subsequent to that instruction. The line predictor may include a memory having multiple entries, each entry storing up to a predefined maximum number of instruction pointers and a fetch address corresponding to the instruction identified by a first one of the instruction pointers. Furthermore, each entry may store additional information regarding the terminating instruction within the entry. *See*, Abstract.

In particular Keller discloses a line predictor entry including a control field 110 that comprises a continuation field (C) 126. *See*, Keller, column 19, lines 32 to 48, and column 21, line 10 to column 22, line 17. The continuation field 126 includes a bit indicative, in one binary state, that the line of instructions crosses a page boundary, and indicative, in the other binary state, that the line of instructions does not cross a page boundary. *See*, Keller, column 22, lines 11-15.

Applicants Traverse the Rejection of Currently Amended Claim 1

Applicants traverse the rejection of currently amended **independent claim 1** (previously claim 2) since the disclosures of Miller and Mohamed do not, in combination, render the claimed invention obvious to one skilled in the art at the time of the invention. More particularly, Claim 1 recites:

> "at least one instruction comprising information that signals *explicitly* how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line, the processing unit being arranged to respond to the information by controlling said part of processing as signaled by the information." (emphasis added).

Claim 1 also recites:

> "the information inserted at compile time explicitly signals whether or not the subsequent memory line has to be prefetched during processing of the instruction, the

processing unit being arranged to start prefetching of the subsequent memory line in response to the information."

Presently pending claim 1 is not obvious over Miller. Miller discloses indicating to the ECU unit and processor where one compressed instruction packet ends and the next compressed instruction packet begins. *See*, Miller, column 5, lines 34-38. However, the information marking instruction packet boundaries does not indicate explicitly whether the "processing is affected by crossing of a boundary to a subsequent memory line". This depends additionally on the position of an instruction packet. Each of Miller's instruction packets will at least have an indicator that indicates the end of the packet. However this packet may be contained within a single address line (e.g. for instruction IP4, Figure 7), or at different address lines (e.g. for instruction IP3, Figure 7). Accordingly, there is no relation between the value of the end of packet indicator and the position of the instruction packet with respect to a subsequent memory line. Let alone that the indicator can indicate "how a part of processing is affected by crossing of a boundary to a subsequent memory line." In particular the indicator does not explicitly indicate whether or not the subsequent memory line has to be prefetched during processing of the instruction. Since Miller neither discloses nor suggests each recited element of claim 1, the claim cannot be obvious over Miller alone.

Moreover, the claimed invention is not rendered obvious by the combined disclosures of Miller and Mohamed. The Final Office Action dated September 22, 2005, asserts that the claimed subject-matter of claim 2, now incorporated in claim 1, would have been obvious to one skilled in the art at the time of the invention in view of the combined disclosures of Miller and Mohamed. Applicants traverse this ground for rejecting the now pending claim 1 (previously claim 2) for the further reasons provided herein below.

As noted above, Miller discloses a system and method for compressing and decompressing variable length instructions stored in a memory 172. The memory 172 is addressed by an addressing unit 190 that is controlled by execution control unit ECU 26. *See*, Miller, column 4, lines 1-2, and column 8, lines 22-24. The compressed instructions comprise a stop bit (or an EP/NEP bit) indicating to the ECU unit and processor where one compressed instruction packet ends and the next compressed instruction packet begins. *See*, Miller column 5, lines 34-38.

Moreover, the portion of Miller at column 11, line 47 to column 12, line 25 describes, with reference to FIG. 8, in more detail how the address of the memory 172 is calculated. From this description it is clear that when reading out and decompressing an instruction packet the addressing system calculates the address in the memory of the next instruction packet. For example when handling instruction packet IP1, the addressing system determines that the next instruction packet is only 16 bits so that the memory address does not need to be incremented. When handling IP2 the addressing system determines that IP3 crosses the 128 bit boundary, and decides to increment the Aleft address.

Mohamed discloses a processor architecture wherein the instructions are not directly fetched from memory 104. Rather, the instructions are fetched indirectly via a buffer 114 or an instruction cache 110. The skilled person aiming to combine the teaching of Mohamed with that disclosed by Miller would similarly insert the instruction cache and prefetch instruction buffer components in the instruction path from the memory 172 to the very long instruction register 180. However, as a consequence of such changes to Miller, the relation between the address of the current instruction and the addresses Aleft, Aright drastically changes. Accordingly the teachings of Miller and Mohamed cannot be simply combined. A combination would require significantly redesigning the addressing architecture of Miller in a way that is neither disclosed nor suggested in the prior art teachings.

Moreover, even if the person of ordinary skill in the art at the time of the invention succeeded in successfully redesigned Miller to enable combining the subject-matter of Miller and Mohamed, this would not result in the subject-matter according to the present invention for at least the following additional reasons. First, according to new claim 1:

> "the information inserted at compile time explicitly signals whether or not the subsequent memory line has to be prefetched during processing of the instruction, the processing unit being arranged to start prefetching of the subsequent memory line in response to the information."

In this regard, Mohamed discloses: "In accordance with the present invention, initially, a compiler sends to decoder unit 102 a "prefetch instruction" flag bit that indicates that a particular block of instructions, or prefetch instructions, are to be placed into prefetch instruction buffer 114, instead of instruction cache 110." *See*, Mohamed, column 3, lines 35-39. Thus, the subject-matter of amended claim 1 differs from the subject-matter disclosed by Mohamed in that the information explicitly signals whether a *subsequent memory line* should be prefetched. The

"subsequent memory line" is defined in claim 1 as the *memory line subsequent to the current memory line and having a boundary therewith.* Hence Applicants' disclosed/claimed subsequent memory line is the memory line *immediately subsequent* to the current memory line.

The claimed "subsequent memory line" is not disclosed in Mohamed. Instead, Mohamed's disclosure teaches away from Applicants' claimed "subsequent memory line" aspect by stating that: "Typically, the prefetch instruction flag bit is decoded *several instructions before the prefetch instructions are decoded,* whether the prefetch instructions are a body of a loop or a power-sensitive function, so that the loop body or the function has been fetched from a memory unit 104 and is accessible for execution without delay." *See,* Mohamed, column 3, lines 55-60.

Second, the subject-matter from amended claim 1 differs in that the information explicitly signals whether *the* subsequent memory line should be prefetched. Hence the information as defined in Claim 1 specifies whether a single (subsequent) memory line should be prefetched. In contrast the prefetch instruction flag bit of Mohamed indicates that a plurality of instructions are to be placed in the prefetch buffer. *See*, Mohamed, claim 1, at column 6, lines 46-49. Accordingly the subject-matter of claim 1 (formerly claim 2) is also non-obvious over the combination of Miller and Mohamed.

Applicants have added new claims 16-19 adding further clarification to claims 1 and 8 in view of the first and second points raised herein above with regard to Applicants' traversal of the rejection of currently amended claim 1 (formerly claim 2) over the prior art.

Applicants submit that currently amended claim 8 is patentable over the prior art for at least the reasons stated herein above with regard to currently amended claim 1. Each of the dependent claims is patentable over the prior art for the reasons set forth for claims 1 and 8 from which each depends.

Applicants Traverse the Rejection of Claim 5 As Obvious Over Miller in view of Keller

The Final Office Action dated September 22, 2005, states that the subject-matter of claim 5, is obvious over the combination of Miller and Keller. Applicants traverse the obviousness rejection of dependent claim 5 for at least for the following additional reasons. The system of Keller typically fetches instructions via an instruction cache. As discussed above, in the architecture of Miller it is essential that the instructions are fetched directly from the instruction memory. Thus, one skilled in the art at the time of the invention, aiming to combine the teaching

of Keller with that disclosed by Miller, would insert the instruction cache in the instruction path from the memory 172 to the very long instruction register 180. However, such combination drastically changes the relation between the address of the current instruction and the Aleft, Aright addresses. Accordingly the teachings of Miller and Keller cannot be simply combined. Such combination of Miller and Kelly would require a significant redesign of at least Miller's disclosed instruction addressing scheme.

Even if the skilled person did succeed in combining the subject-matter of Miller and Keller, the resulting combination would not result in the claimed subject-matter of claim 1. In particular, the continuation field 126 of FIG. 8 of Keller referred to in the Final Office Action has a different function than the information defined in claim 1 of the present invention. The continuation field 126 merely specifies whether the line of instructions crosses a *page* boundary and does not specify whether a *boundary between current and subsequent memory lines* is crossed.

Applicants Traverse the Rejection of Claim 6

The Final Office Action rejected claim 6 as being anticipated by Miller. However, claim 6 specifically recites:

> "the processing unit being a VLIW processing unit containing two or more issue slots for issuing operations from the instruction in parallel to the functional units, the instructions being VLIW instructions, capable of containing two or more operations, the instruction comprising a field distinct from the operations to specify said information."

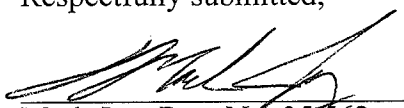
Miller discloses that a token field 112, which may be five (5) bits wide, stores a token that corresponds to an operations ("op") code, a control word, and a form word of an uncompressed instruction. Tokens are selected so that each token corresponds to only one uncompressed instruction in the instruction set. Thus, Miller's token field permits the decompression system to determine both the processing unit that is affected by the instruction as well as the actual instruction for that processing unit. The token field, in effect, both (1) identifies the processing unit and (2) identifies the actual uncompressed instruction. The token may be assigned to uncompressed instructions in any manner, however, the most compression occurs, as described above, when the shortest instruction words are assigned to the most frequently used instructions. *See*, Miller, column 5, lines 39-53. Since the token "corresponds to an operations ("op") code, a

control word, and a form word of an uncompressed instruction", it is definitely not distinct from the operations, as explicitly recited in Applicants' claim 6.

Conclusion

Applicants respectfully submit that the patent application is in condition for allowance. If, in the opinion of the Examiner, a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney.

Respectfully submitted,



Mark Joy, Reg. No. 35,562
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza, Suite 4900
180 North Stetson Avenue
Chicago, Illinois 60601-6731
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: May 19, 2009